



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/614,953	07/08/2003	Young-Gook Ha	3364P116	7127
8791	7590	11/27/2006	EXAMINER	
BLAKELY SOKOLOFF TAYLOR & ZAFMAN			FABER, DAVID	
12400 WILSHIRE BOULEVARD			ART UNIT	PAPER NUMBER
SEVENTH FLOOR				2178
LOS ANGELES, CA 90025-1030				

DATE MAILED: 11/27/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	10/614,953	HA ET AL.
	Examiner	Art Unit
	David Faber	2178

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 23 October 2006.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-15 is/are pending in the application.
 - 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-15 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This office action is in response to the Request for Continued Examination filed 23 October 2006.
2. Claims 1, 6, 8, and 12 have been amended. Claims 13-15 have been added.
3. The rejection of Claims 1, 8, and 12 under 35 USC 112, first paragraph, has been withdrawn necessitated by the amendment. The rejection of Claims 1, 8, and 12 under 35 USC 112, second paragraph, has been withdrawn necessitated by the amendment.
4. Claims 1-15 are pending. Claims 1, 8, and 12 are independent claims.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.
6. Claims 1-15 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.
7. As per independent claim 1, Claim 1 recites the limitation "document components" in line 10. It is unclear if the element "document components" within claim 1 is depending on the "document components" in claim 1, line 7 or is a new element. Thus, there is insufficient antecedent basis for this limitation in the claim.
8. As per independent claim 1, Claim 1 recites the limitation "context processor" in line 13. It is unclear if the element "context processor" within claim 1 is depending on

Art Unit: 2178

the "context processor" in claim 1, line 11 or is a new element. Thus, there is insufficient antecedent basis for this limitation in the claim.

9. As per dependent claim 2, Claim 2 recites the limitation "a graphic user interface" in line 10. It is unclear if the element "a graphic user interface" within claim 1 is depending on the "a graphic user interface" in claim 1, line 4 or is a new element. Thus, there is insufficient antecedent basis for this limitation in the claim.

10. As per dependent claim 2, Claim 2 recites the limitation "assembly rules" in line 14. It is unclear if the element "assembly rules" within claim 2 is depending on the "assembly rules" in claim 1, line 5 or is a new element. Thus, there is insufficient antecedent basis for this limitation in the claim.

11. As per dependent claim 4, Claim 4 recites the limitation "document components" in page 4, line 2, and line 8. It is unclear if the element "document components" within claim 2 is depending on the "document components" in claim 1, line 5 or is a new element. Thus, there is insufficient antecedent basis for this limitation in the claim.

12. As per dependent claim 6, Claim 6 recites the limitation "assembly rules" in line 3. It is unclear if the element "assembly rules" within claim 6 is depending on the "assembly rules" in claim 1, line 5 or is a new element. Thus, there is insufficient antecedent basis for this limitation in the claim.

13. As per dependent claim 6, Claim 6 recites the limitation "context rules" in line 9. It is unclear if the element "context rules" within claim 6 is depending on the "context rules" in claim 1, line 5 or is a new element. Thus, there is insufficient antecedent basis for this limitation in the claim.

Art Unit: 2178

14. Claim 6 and 7 recites the limitation "the grammar neutral document objects" in line 12 and line 3-4, respectively. There is insufficient antecedent basis for this limitation in the claim.

15. As per independent claim 9, Claim 9 recites the limitation "a graphic user interface" in page 6, lines 4. It is unclear if the element "a graphic user interface" within claim 1 is depending on the "a graphic user interface" in claim 8, lines 5-6 or is a new element. Thus, there is insufficient antecedent basis for this limitation in the claim.

16. As per dependent claim 10, Claim 10 recites the limitation "assembly rules" in line 2. It is unclear if the element "assembly rules" within claim 10 is depending on the "assembly rules" in claim 8, line 6 or is a new element. Thus, there is insufficient antecedent basis for this limitation in the claim.

17. As per dependent claim 10, Claim 10 recites the limitation "context rules" in line 6. It is unclear if the element "context rules" within claim 10 is depending on the "context rules" in claim 8, line 6-7 or is a new element. Thus, there is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 103

18. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Art Unit: 2178

19. Claims 1-12 remain rejected under 35 U.S.C. 103(a) as being unpatentable over Cornelia et al (US Patent #6,065,026, patented 5/16/2000) further in view of Person et al (Person et al, "Special Edition Using Microsoft Word 97", published 12/16/1996, pp 1-15, 16-20) in further view of Meyer, (Meyer, "aTool – Creating Validated XML Documents on the Fly Using MS Word," published 10/20/2002, pp 113-121).

As per independent Claim 1, Cornelia et al discloses a system comprising:

- A document generation rule formulator that a user employs to designate document rules through a graphic user interface (Col 6, line 66 – Col 7, line 3: User is able to designate rules to components that are used to create documents through a graphic user interface; a tree viewer. When the user selects which components are to be included into the document, the user is designating document generation rules (i.e. assembly rules) of which components are to be assembled into the document. Thus the assembly rules disclose what components are to be assembled into the document.)
- a document component library for storing and managing document component summary information and document components that represent specific concepts; (Column 2, line 66 – Column 3, line 2: discloses a library used for storing components which are used to be assembled to create new documents. In addition, Column 9, lines 44-49, discloses Find Component menu option that is able to display a component dialog disclosing the component's name, description, author, text content, etc. Since the component contains all this information and displays it, the library stores

components that contain summary information which represent specific concepts.)

- a component assembler for processing the assembly rules and assembling document components from the document component library. (Column 20, lines 60 – 65: discloses a component assembler of creating documents by dragging and dropping language component icons where each icon represents a component. The components are stored in the library. (Col 2, line 66 – Col 3, line 2) Therefore, since the user is designating assembly rules by choosing which components are included in the document, the documents are assembled from the components chosen (assembly rules) from the library.)
- Creating a grammar neutral document object from the assembled document components (Column 20, line 64 – Column 21, line 4: Once the components have been assembled, a document is created.)

However, Cornelia et al fails to specifically disclose the document generation rules include context rules; and a context processor for processing context rules. However, Person et al discloses Word contains templates, which contain parts of a document and features used for a specific type of document. Person et al's discloses where context conditions are used by formulation rules where the template created allows the user to easily enter information onto ASK or FILLIN field by requiring only the user just to point, click, and type to fill out a form. (Page 4, Paragraph 4; FIG 6.4) Since the user is only required to fill out the ASK or FILLIN fields on the form, then context

Art Unit: 2178

rules are applied. This process acts as a context processor that discloses an embodiment using a template showing context conditions allowing the user only have to point, click, and type information in a already constructed form by the template's context rules.

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to have modified Cornelia et al's method with Person et al's disclosure since Person et al's disclosure of using a Microsoft Word template is a tool one could have used to further increase efficiency, productivity, and consistency while reducing company's costs when creating a document.

In addition, Cornelia et al and Person et al fail to specifically disclose a document grammar connector for converting the grammar neutral document object, which is suitable for processing in a program of a computer system, into a grammar-connected document that is in a human-readable string form used in an actual business. However, Meyer discloses using a tool extension to Microsoft Word (Abstract, lines 1-2, pg 1) that would create validated XML documents using Microsoft Word. (Title, pg 1) Meyer's tool would convert the MS Word document into XML generating a grammar-connected document (Page 116, Right Column, lines 30-37) by making sure the document is valid and complaint with its DTD (Page 114, Right Column, Lines 30-31). This process acts as a document grammar connector. An XML document is inherently considered as a recognizable string format by the user.

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to have combined Cornelia et al's and Person et al's methods with

Meyer's aTool method since it would have provided the aTool as a hybrid solution that offers the benefits of MS Word with the costs of a little less XML support.

As per dependent Claim 2, Cornelia et al discloses:

- a component selector for displaying usable component items that are provided by a corresponding library based on document component summary information searched in the document component library, the document component summary information including at least a component ID, a component name, and a component type, and optionally including various different types of information that represent other components; (Column 9, lines 42-52: Discloses able to using a Find Component menu option to find components based on the component's information such as the component's name in which the system returns with a list of components in the library based on the component's information searched. The user is able to insert components into the document from the component list.)
- a document component assembler for forming an area where component structures are modeled based on user input through a graphic user interface, the user dragging the needed document components appearing in the component selector and dropping the documents at a suitable location in the document component assembler to thereby generate document structures such structures are formulated as assembly rules (Column 20, line 60 – Column 21, line 4: Discloses documents being created by dragging and dropping components into a list for the document using a tree viewer. Once

Art Unit: 2178

the list been created, the word document with complete content is generated by a user action. This creation creates a structured document containing components placed in a structural manner. Therefore, since the user is designating assembly rules by choosing which components are included in the document, the documents are assembled from the components chosen (assembly rules) from the library.)

However, Cornelia et al fails to specifically disclose the assembly rules include IDs of all document components and structural information between each component; and disclose a context condition compiler for forming an area where context conditions realized through pairs of conditions and actions are compiled to enable insertion into document structures, the context condition compiler enabling the formulation of context rules, which allow the processing of actions, in the document generation rule processor in the case where conditions are satisfied for a specific business context during document assembly.

Cornelia et al discloses the author has the ability to get a listing of all the components in the document that displays component identifiers such as the component's name. (Column 11, lines 51-58) Since Cornelia et al's application is built using Microsoft Word (Column 6, lines 22-47), Cornelia discloses the ability of Word being able to view the assembled document with the components and the structured order of how the document components appear (Column 11, lines 52-65). Therefore, Word would been able to include the identifiers using the Show Document Components menu option from Cornelia et al's application since it was built using Microsoft Word.

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to have combined Cornelia et al's with the use of identifiers since it would have allowed using computers to assemble documents to reduce the amount of time that attorneys and other individuals who prepare long legal or transactional documents spend on the mechanics of document preparation.

Furthermore, Person et al discloses Word contains templates, which contain parts of a document and features used for a specific type of document. (Page 1) Person et al's discloses where context conditions are used by the formulation rules where the template created allowing the user to easily enter information by requiring only the user just to point, click, and type to fill out a form. (Page 4, Paragraph 4; FIG 6.4)

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to have combined Cornelia et al's method with Person et al's disclosure since Person et al's disclosure of using a Microsoft Word template is a tool one could have used to further increase efficiency, productivity, and consistency while reducing company's costs when creating a document.

As per dependent Claim 3, Cornelia et al fails to specifically disclose the assembly rules and the context rules are output as a single document generation rule. However, Person et al discloses Word contains templates, which contain parts of a document and features used for a specific type of document. (Page 1) Person et al discloses a specific type of a form template, where assembly and context rules are used together in which the assembly rules are used create a table that creates a form while the context rules are used for font size and font type for the text. (FIG 6.2) thus a

template is a single document generation rule involving the combination of assembly and context rules.

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to have combined Cornelia et al's method with Person et al's disclosure of templates since Person et al's disclosure of using a Microsoft Word template is a tool one could have used to further increase efficiency, productivity, and consistency while reducing company's costs when creating a document.

As per dependent Claim 4, Cornelia et al discloses a system:

- the document component summary information for recording various document components that constitute business documents and detailed information on all components included in a present library; and ; (FIG 26; Column 12, line 65 – Column 13, line 1: Discloses the ability to view one component from the library which discloses detailed component information that is used for business purposes (Column 6, lines 43-47)
- a component library interface for connection to external modules, (FIG. 3; Column 5, lines 27-38: Discloses the word processor, as a separate module, having an API that interacts with the library object which interacts with the library.)
- the document generation rule formulator searches the document component summary information through the component library interface, and the document generation rule processor uses document component Ids to accumulate document components required for document assembly. (Column

6, lines 6-12 discloses the interaction using the library. This disclosure enables using a Find Component menu option to find components based on the component's information such as the component's name in which the system returns with a list of components in the library based on the component's information searched then allowing the author to insert the component into document thus outputting onto the document. (Column 9, lines 42-52))

However, Cornelia et al fails to specifically discloses the component Ids are numbers specific to each component. However, it was well-known to one of ordinary skill in the art at the time of applicant's invention that a number can be a name and that identifiers were programmed as/into numbers within a data structure .

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to have combined Cornelia et al's method with the disclosure of numbers used as name as identifiers since it would have allowed a user an easier method for storing data within a recording medium.

As per dependent Claim 5, Cornelia et al discloses a system:

- wherein the document components stored in the document component library include simple components of a single type and complex components realized through a structure of a plurality of simple components. (Column 2, line 66 – Column 3, line 2 discloses a library containing components which are unit of text that is shared among documents. A component may contain

variable text in order to facilitate flexibility and foster re-use. (Column 2, lines 48-50) Column 10, lines 8-34 discloses adding variable text into the component.)

As per dependent Claim 6, Claim 6 recites similar limitations as in Claim 1, and is similarly rejected under rationale. Cornelia et al discloses a system:

- a component assembler is to read assembly rules into the document generation rules and use document component IDs to accumulate from the document component library the document components required in the assembly rule, resulting assembled components are outputted. (Column 2, line 66 – Column 3, line 2; Column 9, lines 42-52: Discloses using a Find Component menu option to find components based on the component's information such as the component's name in which the system returns with a list of components in the library based on the component's information searched then allowing the author to insert the component into document thus outputting onto the document.)

In addition, Cornelia discloses the ability to assemble a document by choosing the components, and be able to change the order of the components listed in the document. (Col 14, lines 14-15) However, Cornelia et al fails to specifically disclose assembling the document components using structural information between components. However, Person et al discloses that Microsoft Word contains structural information between each component when present such as the paragraph markings

shown in FIG 10.3. Between each component, a paragraph mark is shown thus showing structural information is used to separate the components. (Pages 16-20) Therefore, since Cornelia et al's application is built off of Microsoft Word, Word would have been able to determine the structural information between each component when components are assembled.

It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to have modified Cornelia et al's method of document creating using components with Person et al's disclosure of Word's detection of structural information since it would provided the benefit of identifying the components from each other when just viewing the document without any other functions.

In addition, Cornelia et al fails to specifically disclose a context processor for reading context rules in the document generation rules, and if a specific business context satisfies the conditions of the context rules, applying designated actions to the assembled components to thereby ultimately generate the grammar neutral document objects.

Furthermore, Person et al's discloses where context conditions are used by the formulation rules where the template created allowing the user to easily enter information by requiring only the user just to point, click, and type to fill out a form. (Page 4, Paragraph 4; FIG 6.4) This process acts as a context processor that discloses an embodiment using a template showing context conditions allowing the user only have to point, click, and type information in a already constructed form by the template's assembly and context rules.

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to have combined Cornelia et al's method with Person et al's disclosure since Person et al's disclosure of using a Microsoft Word template is a tool one could have used to further increase efficiency, productivity, and consistency while reducing company's costs when creating a document.

As per dependent Claim 7, Cornelia et al discloses a document output unit saving final documents to the library after creating a document with components (Column 15, lines 49-57) for business uses (Column 6, lines 43-47) However, Cornelia et al and Person et al fail to specifically disclose a grammar converter supporting grammar for specific business systems and converting the grammar neutral document objects into grammar-connected document objects;

However, Meyer discloses using a tool extension to Microsoft Word that would create validate XML documents using Microsoft Word. Meyer's tool would convert the MS Word document into XML generating a grammar-connected document (Page 116, Right Column, lines 30-37) by making sure the document is valid and in compliance with its DTD (Page 114, Right Column, Lines 30-31). An XML document is inherently considered as a recognizable string format by the user.

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to have combined Cornelia et al's and Person et al's methods document generation with Meyer's aTool method to create XML document from Word

that would have provided the aTool as a hybrid solution that offers the benefits of MS Word with the costs of a little less XML support.

As per independent Claim 8, Claim 8 recites similar limitations as in Claim 1 and 6, and is similarly rejection under rationale. Furthermore Cornelia et al discloses a method comprising:

- (a) storing document component summary information and document components that represent specific concepts; (Column 2, line 66 – Column 3, lines 1-2: discloses a library storing components used to be assembled to create new documents. In addition, Column 9, lines 44-49, discloses a option within Cornelia et al's application of a Find Component function that able to display a component dialog disclosing the component's name, description, author, text content etc. Since the component contains all this information and displays it, the library stores components that contain summary information which represent specific concepts.)
- (c) accumulating document components needed for document assembly from a document component library according to the assembly rules, and generating grammar neutral document objects (Column 20, lines 60 – Col 3, line 2: discloses creating documents by dragging and dropping language component icons where each icon represents a component. Documents are created by the dragging and dropping components into a list for the document thus creating non-grammar objects within a document. When the user selects

which components are to included into the document, the user is using assembly rules of which components are to be assembled into the document, which the rules are designated based on the user's selection.)

In addition, Cornelia et al and Person et al fail to specifically disclose converting the grammar neutral document objects, which are suitable for processing in a program of a computer system, into grammar-connected documents in a human-readable string form used in an actual business. However, Meyer discloses using a tool extension to Microsoft Word that would create validate XML documents using Microsoft Word. Meyer's tool would convert the MS Word document into XML generating a grammar-connected document (Page 116, Right Column, lines 30-37) by making sure the document is valid and compliance with its DTD (Page 114, Right Column, Lines 30-31). An XML document is inherently considered as a recognizable string format by the user.

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to have combined Cornelia et al's and Person et al's methods of document generation with Meyer's aTool method of creating XML documents from Microsoft Word would have provided the aTool as a hybrid solution that offers the benefits of MS Word with the costs of a little less XML support.

As per dependent Claim 9, Cornelia et al discloses a method:

- displaying a list of usable components provided library based on the by a corresponding document component summary information searched in the document component library; (Column 9, lines 42-52: Discloses using a Find

Component menu option to find components based on the component's information in which the system returns with a list of components in the library based on the component's information searched.)

- dragging required documents appearing in a component selector and dropping the documents at a suitable location in a document component assembler, which forms an area where component structures are modeled based on user input through a graphic user interface, to thereby generate document structures; and (Column 20, line 60 – Column 21, line 4: Discloses documents being created by dragging and dropping components into a list for the document using a tree viewer. Once the list been created, the word document with complete content is generated by a user action. This creation creates a structured document containing components placed in a structural manner.)

However, Cornelia et al fails to specifically disclose compiling context conditions realized through pairs of conditions and actions, and allowing insertion of the context conditions into document structures. However, Person et al discloses Word contains templates, which contain parts of a document and features used for a specific type of document. (Page 1) Person et al's discloses where context conditions are used by the formulation rules where the template created allowing the user to easily enter information by requiring only the user just to point, click, and type to fill out a form. (Page 4, Paragraph 4; FIG 6.4)

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to have combined Cornelia et al's method with Person et al's disclosure of templates since Person et al's disclosure of using a Microsoft Word template is a tool one could have used to further increase efficiency, productivity, and consistency while reducing company's costs when creating a document.

As per dependent claim 10, Claim 10 recites similar limitations as in Claim 6, and is similarly rejected under rationale.

As per dependent claim 11, Claim 11 recites similar limitations as in Claim 7, and is similarly rejected under rationale.

As per dependent claim 12, Claim 12 recites similar limitations as in Claim 8, and is similarly rejected under rationale.

20. Claims 13-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cornelia et al (US Patent #6,065,026, patented 5/16/2000) further in view of Person et al (Person et al, "Special Edition Using Microsoft Word 97", published 12/16/1996, pp 1-15, 16-20) in further view of Meyer, (Meyer, "aTool – Creating Validated XML Documents on the Fly Using MS Word," published 10/20/2002, pp 113-121) in further view of Hughes ("Stoking the AbiWord Fire", published 2/4/2002, pp 1-2).

Cornelia et al discloses that the document created can support a variety of word processor formats; however, Cornelia et al, Person et al, and Meyer fail to disclose the grammar neutral document object(s) are Extensible Markup Language documents.

However, Hughes discloses a word processor, Abiword, which its word processor format is in XML. (pg 2, paragraph 2)

It would have obvious to one of ordinary skill in the art at the time of Applicant's invention to have modified Cornelia's invention with Abiword's word processor format since it would have provided the benefit of word processors gaining in portability and compatibility to exchange data between different applications.

Response to Arguments

21. Applicant's arguments filed 23 October 2006 have been fully considered but they are not persuasive.

22. In regards to pages 9-10, Applicant argues in regards to Claim 1 that Cornelia et al produces a complete document and that does not follow Applicant's invention wherein a generation rule processor creates a grammar neutral document object, which is further processed by a document grammar connector for converting the grammar neutral document object into a grammar-connected document that is in a human-readable string form for use in an actual business. However, the Examiner disagrees.

The Examiner would like to point out arguing that Cornelia's output is a completed document is vague and indefinite since the Applicant did not define in the argument what is considered as a completed document. According to Applicant's explanation, the Examiner believes, based on the specification, a completed document is the result of a grammar neutral document object being converted into a grammar-

connected document that is in a human-readable string form for use in an actual business, wherein the grammar-connected document is believed to be a completed document. Furthermore, Applicant discloses that a grammar neutral document object is just an object without any grammar (e.g. XML Schema, XML DTD) since it needs to go through a grammar connector to have grammar as disclosed in page 12 of the specification. Therefore, Cornelia discloses creating a document from components which is considered an uncompleted document since it does not contain any grammar (as defined in the specification). Thus, with the use of Meyer and the rationale incorporated, Meyer fulfills the deficiencies of Cornelia et al and Person et al to create a grammar-connected document.

23. In regards to page 10, Paragraph 2, Applicant argues that Cornelia et al and Person et al fails to teach or suggest a component assembler and context processor that operate as required by Claim 6. However, the Examiner disagrees.

Cornelia et al discloses a system a component assembler is to read assembly rules into the document generation rules and use document component IDs to accumulate from the document component library the document components required in the assembly rule, resulting assembled components are outputted. (Column 2, line 66 – Column 3, line 2; Column 9, lines 42-52: Discloses using a Find Component menu option to find components based on the component's information such as the component's name in which the system returns with a list of components in the library

based on the component's information searched then allowing the author to insert the component into document thus outputting onto the document.)

In addition, Cornelia discloses the ability to assemble a document by choosing the components, and be able to change the order of the components listed in the document. (Col 14, lines 14-15) However, Cornelia et al fails to specifically disclose assembling the document components using structural information between components. However, Person et al discloses that Microsoft Word contains structural information between each component when present such as the paragraph markings shown in FIG 10.3. Between each component, a paragraph mark is shown thus showing structural information is used to separate the components. (Pages 16-20) Therefore, since Cornelia et al's application is built off of Microsoft Word; Word would been able to determine the structural information between each component when components are assembled.

It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to have modified Cornelia et al's method of document creating using components with Person et al's disclosure of Word's detection of structural information since it would provided the benefit of identifying the components from each other when just viewing the document without any other functions.

In addition, Cornelia et al fails to specifically disclose a context processor for reading context rules in the document generation rules, and if a specific business context satisfies the conditions of the context rules, applying designated actions to the

assembled components to thereby ultimately generate the grammar neutral document objects.

Furthermore, Person et al's discloses where context conditions are used by the formulation rules where the template created allowing the user to easily enter information by requiring only the user just to point, click, and type to fill out a form. (Page 4, Paragraph 4; FIG 6.4) This process acts as a context processor that discloses an embodiment using a template showing context conditions allowing the user only have to point, click, and type information in a already constructed form by the template's assembly and context rules.

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to have combined Cornelia et al's method with Person et al's disclosure since Person et al's disclosure of using a Microsoft Word template is a tool one could have used to further increase efficiency, productivity, and consistency while reducing company's costs when creating a document.

24. In regards to page 11, Paragraph 1, Applicant argues that the characterization of Meyer is reasonably accurate, but the direction of conversion is wrong. Applicant goes on to say that grammar-neutral document objects (which may be XML as recited in claim 14) are converted into grammar-connected documents....", wherein stating it is almost of the reverse of Meyer's teaching. However, the Examiner disagrees.

Claims 1, 7, 8, and 12, which discloses the limitation of wherein converts grammar-neutral document objects into grammar-connect documents.... fail to state

within those claims of the feature that the grammar-neutral document objects are in XML; therefore grammar neutral document objects are viewed as to be in any (text) format, such as the Microsoft Word processor (text) format used by Cornelia in further view of Person. Thus, with the use of Meyer, and the rationale incorporated, Meyer converts a Microsoft Word document into an XML document, a grammar-connected documents thus using Meyer in the correct direction and functionality.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David Faber whose telephone number is 571-272-2751. The examiner can normally be reached on M-F from 8am to 430pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Stephen Hong, can be reached on 571-272-4124. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

David Faber

Application/Control Number: 10/614,953

Art Unit: 2178

Patient Examiner
AU 2178

DF

Page 25



STEPHEN HONG
SUPERVISORY PATENT EXAMINER